


Network arrangement for glassware forming system.

Patent Number: ☐ EP0667693, A3
Publication date: 1995-08-16
Inventor(s): HWANG DAVID K (US)
Applicant(s):: EMHART GLASS MACH INVEST (US)
Requested Patent: ☐ JP7267653
Application Number: EP19950300912 19950214
Priority Number(s): US19940196307 19940215
IPC Classification: H04L12/00 ; G05B19/00 ; C03B9/00
EC Classification: C03B9/41, H04L29/06E
Equivalents: ☐ US5475601

Abstract

A glassware forming machine comprising at least one event driven database operating on one protocol, at least one workstation operating on a second protocol including a library of API calls in the one protocol, a socket library for sending call packets in the second protocol, a gateway library for transporting an API call in said one protocol to a call packet in the second protocol, a network gateway including a socket library for receiving call packets in the second protocol, and a gateway service for translating a call packet received by the network gateway socket library in the second protocol to an API call in said first protocol and for delivering the translation call to the event driven data base. 

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-267653

(43) 公開日 平成7年(1995)10月17日

(51) Int.Cl. ⁴	願別記号	庁内整理番号	F I	技術表示箇所
C O S B 9/41				
11/16				
G O S B 19/02	D			

審査請求 未請求 請求項の数 2 O L (全 11 頁)

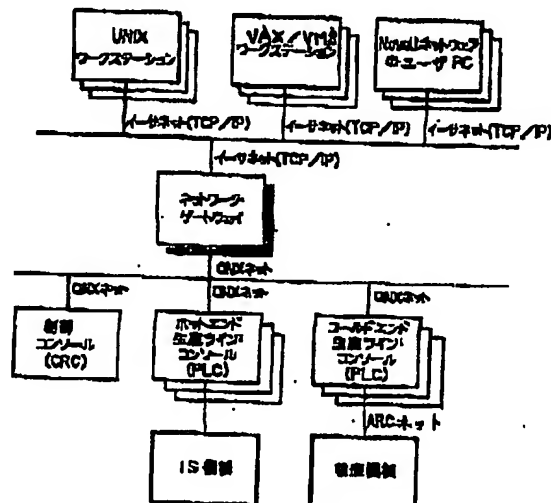
(21) 出願番号	特開平7-27007	(71) 出願人	581255003 エムハート・グラス・マシーナリー・イン ベストメンツ・インコーポレーテッド EMHART GLASS MACHIN ERY INVESTMENTS INC ORPORATED アメリカ合衆国デラウェア州19889, ウィ ルミントン, ワン・ロドニー・スクエアー (番地なし), アールエルエフ・サービ ス・センター
(22) 出願日	平成7年(1995)2月15日	(72) 発明者	デーヴィッド・ケイ・ワン アメリカ合衆国コネチカット州06268, ス トアーズ, ティンバー・ドライブ 37
(31) 優先権主張番号	196307	(74) 代理人	弁理士 湯淺 義三 (外6名)
(32) 優先日	1994年2月15日		
(33) 優先権主張国	米国 (US)		

(54) 【発明の名称】 ガラス製品等の形成システム

(57) 【要約】

【目的】 遠隔のユーザ・ワークステーションからガラス形成及び検査マシンのデータベースへのアクセスを可能にする。

【構成】 個別のガラス製品形成機械は、イベント駆動型のデータベースを含む複数の生産ライン・コンソール (PLC) を有しており、該イベント駆動型データベースのための保管データが、制御コンソール (CPC) に記憶されている。UNIX、VAX/VMS、又はNovellのネットワークを実行するワークステーションから、イーサネットとネットワーク・ゲートウェイを介して、QNXネットワークのデータベースにアクセスできる。また、コールドエンドPLCは、ARCネット及びQNXネットの両方とインターフェースし、検査機械からデータを収集、ネットワーク・ゲートウェイを介してワークステーションに転送する。検査制御ソフトウェアは、コールドエンドPLC上で実行される。



(2)

特開平7-267653

1

【特許請求の範囲】

【請求項1】 ガラス製品等の形成システム

(a) 第1のプロトコルの上で動作する少なくとも1つのイベント駆動型データベースと、

(b) 第2のプロトコルの上で動作する少なくとも1つのワークステーションであって、

前記第1のプロトコルにおけるAPIコールのライブラリと、

前記第2のプロトコルにおけるコール・バケットを送るソケット・ライブラリと、

前記第1のプロトコルにおけるAPIコールを前記第2のプロトコルにおけるコール・バケットに翻訳するゲートウェイ・ライブラリと、を含むワークステーションとを含むワークステーションと、

(c) ネットワーク・ゲートウェイであって、

前記第2のプロトコルにおけるコール・バケットを受け取るソケット・ライブラリと、

前記第2のプロトコルにおける前記ネットワーク・ゲートウェイのソケット・ライブラリを、前記第1のプロトコルにおけるAPIコールに翻訳し、前記翻訳コールを前記イベント駆動型データベースに搬送するネットワーク・ゲートウェイとを含むネットワーク・ゲートウェイとを備えていることを特徴とするガラス製品等の形成システム。

【請求項2】 請求項1記載のガラス製品等の形成システムにおいて、

前記ゲートウェイ・サービスはさらに、前記イベント駆動型データベースから受信された前記第1のプロトコルにおけるデータを、前記第2のプロトコルにおけるデータに翻訳する手段を含み、

前記ネットワーク・ゲートウェイはさらに、前記第2のプロトコルにおけるデータのバケットを前記ワークステーションのソケット・ライブラリに搬送する手段を含み、

前記ゲートウェイ・ライブラリはさらに、前記第2のプロトコルにおける前記搬送されたデータを前記第1のプロトコルを有するデータに翻訳する手段を含んでいることを特徴とするガラス製品等の形成システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、ガラス製品等の形成システムに関し、特に、遠隔地のユーザのワークステーションからガラス形成及び検査マシンのデータベースへのアクセスを提供するネットワーク・ゲートウェイを備えたガラス製品等の形成システムに関する。

【0002】

【従来の技術】 ガラス製品工場には多くのコンピュータ制御されたマシンがあり、これらのマシンは共同してガラス製品形成プロセスを固定している。これらのマシンには、たとえば、米国特許第4641269号に

2

開示されたIS（個別セクション）機械すなわちマシンが含まれ、その複数のセクションはそれぞれが独立のコントローラによって制御されている。これらのコントローラのための保管データは、制御室のコンソールに集中して記憶されている。形成されたビン等のガラス製品は、それぞれが自己の独立のコントローラを有する1つ又は複数のディスクリット・マシンを有する検査装置によって検査される。

【0003】

10 【発明が解決しようとする課題】 本出願の被譲渡人が販売するこのようなISマシンや検査装置は、非標準的なQNXプラットフォーム（クアントム・ソフトウェア・システムズ社が発行出した、マイクロコンピュータ・システムのためのマルチタスク・マルチユーザ・リアルタイムのオペレーティング・システム）の基で動作する。この非標準的なプラットフォームは、標準的なイーサネット（TCP/IP）のネットワーク環境におけるUNIXやVAX/VMSやノベル社のネットワークなどの標準的なプラットフォーム上で動作している外部のコンピュータからはアクセスできず、従って、ガラス製品工場におけるユーザ・ワークステーションが、これらのISマシンや検査装置のデータベースにアクセスできないという問題点があった。したがって、本発明の目的は、遠隔地のユーザ・ワークステーションからこれらのガラス製品形成及び検査マシンのデータベースへのアクセスを提供するネットワーク・ゲートウェイを与えることができるようにすることである。

【0004】

30 【発明の概要】 本発明においては、上記した目的を、ネットワーク環境内に分散しているデータベースの地理的な位置を隠してユーザ・ワークステーションからアクセスされた際にはローカルであるかのように見せる分散型のコンピュータ環境によって達成している。ゲートウェイは、機械（マシン）のデータベースと遠隔のワークステーションとの間の2方向データ通信を提供する。遠隔のユーザがこのゲートウェイを介して必要とするデータを受信することができるだけでなく、データベースが何らかの自発的変更（unsolicited changes）を有する場合には、このゲートウェイは、その遠隔のユーザに直ちにそれを通信することができるように構成されている。本発明のこれ以外の目的及び効果は、本明細書の以下の記述と特許法の命じるところに従い本発明の原理を組み入れた好適実施例を図解する添付の図面とから明らかになる。

【0005】

40 【実施例】 図1には、ネットワーク・ゲートウェイと、ローカル・コンピュータと、外部のアプリケーション装置との関係が示されている。ガラス製品を形成するIS（個別セクション）機械は、複数の生産ライン・コンソールを有しており、各コンソールは、当該IS機械への

(3)

特開平7-267653

3

すべての管理制御情報をサポートするイベント駆動型のデータベースを含んでいる。複数の機械のこれらすべてのイベント駆動型データベースのための保存 (archive) データは、制御コンソールのハード・ドライブ (ディスク) データベースに集中的に記憶されている。

また、QNXネットワーク上で利用可能なのは、ARCネットワークを介して検査装置データベースにアクセスする検査接続ソフトウェアを実行するコールドエンド生産コンソールである。本発明の最終目標は、ユーザのアプリケーションが、標準的なイーサネットTCP/IPネットワーク環境においてUNIX、VAX/VMS、又はNovell社のネットワークウェアのいずれかを走らせているワークステーションから、ネットワーク・ゲートウェイを介してこれらのQNX型データベースにアクセスすることを可能にすることである。

【0006】ネットワーク・ゲートウェイは、IBM-PCコンパチブルなコンピュータであり、以下のハードウェア及びソフトウェア要素から成る。すなわち、

- ・8MBのメモリを有する486/33MHzのIBM-PC (ハード・ドライブなし)

- ・V2.15F又はそれ以降のQNXオペレーティング・システム

- ・QNXネット・ボード

- ・パークレー・ソケット・ライブラリ及びCMC-640イーサネット・インターフェース・カードを含むQNXデバイス・ドライバのためのFastech社のTCP/IP接続

- ・データベースへのアクセスのためのゲートウェイ・サーバ・ソフトウェア

これによって、ワークステーションは、イーサネットを介して、QNXネットワーク上の要求されたデータベース情報を収集することが可能になる。ただし、ゲートウェイ・トラフィック (伝送路) は、双方向性であり、すなわち、ネットワーク・ゲートウェイもワークステーションへのデータ転送を開始することができる。実際には、開始されたイベントは、そのワークステーションによって定義され要求されたデータベースのアイテムだけに限定される。

【0007】ネットワーク・ゲートウェイのソフトウェアは、1つのユーザ (クライアント) ・ワークステーション内の複数の処理又は複数のユーザ・ワークステーションの間の1つ又は複数の処理の中に存在する複数のユーザ (クライアント) ・ポートをサポートする。ネットワーク・ゲートウェイ・ソフトウェア自体は、1つのCPU又はそれぞれが別の名前を有する複数のPCの中にも存在することができる。コールドエンド生産ライン・コンソール (PLC) は、ARCネットワーク (ARCネット) 及びQNXネットワーク (QNXネット) の両方とインターフェースするIBM-PCコンパチブル・コンピュータである。ARCネットワークに対するAR

4

Cネット・インターフェースは、すべての検査装置からデータを集め、他方で、QNXネットワークは、コールドエンドPLCを、QNXネットワーク上のノードにする。検査接続ソフトウェアは、コールドエンドPLC上でQNXの下に実行される。

【0008】このシステムを構築する別の方法は、単にARCネット・カードをゲートウェイPCに挿入し、かつ検査接続及びゲートウェイ・サービス・ソフトウェアを同じPCで実行、すなわち走らせることによって、コールドエンドPLCとネットワーク・ゲートウェイ・ハードウェアとを1つのPCボックス内に組み合わせることである。ある限界までは、このアプローチは、コストを減少させ、効率を向上させるという両方の点で効果を有する。しかし、システムに2つ以上のコールドエンド検査機械が付加された場合には、検査接続を走らせる別個のPCを有するという通常の構成では、ユーザに、増加する要求に答えるだけの付加的なPCを加えるフレキシビリティを与えることになる。

【0009】ゲートウェイ・ソフトウェアは、クライアントすなわちユーザ側 (ワークステーション) とサーバ側 (ゲートウェイ) の部分に分けられる。クライアントのソフトウェアは、C言語によるソース・コード・フォーマットのAPIライブラリ・ファイルの組を含み、それにより、ユーザは、コンパイルすなわち編纂を行ってそれを自分自身のアプリケーション・コード及びTCP/IPソケット・ライブラリにリンクさせることができる。QNXの下でゲートウェイ上を走っているソフトウェアは、クライアントからデータベースへのデータ取得要求と、データベースからユーザ (クライアント) ・アプリケーションへのデータベース変更イベントとを与える。サーバ・ソフトウェアである。サーバ・ソフトウェアは、複数のセッションに加えて複数のクライアント接続をサポートする。

【0010】QNXアプリケーションの開発者に対するものとして、本出願の被譲渡人は、イベント駆動型データベース (EDDB) のツール・キットを公知にしており、これは、QNX環境におけるEDDBデータベース・アクセスのための標準的 (ネイティブ) なEDDBライブラリAPI (図2) を含んでいる。QNX環境においては、アプリケーション・タスクは、ローカルなEDDBライブラリからEDDB・APIコールを生じて、ローカル又は遠隔地のEDDBデータベースをオープン (位置指定) にし、その後、そこからのデータにアクセスすることができる。このようなネットワーク本来の環境においては、ユーザは、データベースの位置を知っている必要はない。EDDBデータベースの地理的位置は、QNXネットワークの設備によって明確に指定される。

【0011】ゲートウェイは、EDDB/IC (検査接続) のライブラリAPIをネットワークを横断して延長

(4)

特開平7-267663

5

しており、該EDDB/ICのプログラミング・インターフェースを用いて、開発者が、ユーザ・アプリケーション・プログラムをUNIX、VAX/VMS/Novellワークステーション上で書くことができるようにする。このために、クライアントのノード上のEDDB/ICライブラリは、その下にソフトウェアの層を付加することによって、ユーザすなわちクライアントとゲートウェイとの間のパケット転送をサポートするように強化されなければならない。また、ソフトウェアの対応する組が、ゲートウェイのノード上に存在してその要求に応じなければならない。図3は、このアプローチによる分散型のソフトウェア構造を示している。すなわち、ソフトウェアの一方の部分であるゲートウェイ・ライブラリがユーザのプラットフォーム上に分散しており、他方の部分であるゲートウェイ・サービスはゲートウェイ上に存在し、その両者が、バークレー・ソケット (Berkeley Sockets) ・ライブラリ (BSD 4.3) を用いて、低レベルのTCPソケット・セットアップと、セッション接続と、パケット転送タスクとを実行する。

【0012】EDDB/ICコールを行うことによって、UNIX/VMS/Novellアプリケーション・プログラムは、遠隔地のデータベースにそれらがあたかもローカルに利用可能であるかのようにアクセスすることができる。ユーザ・アプリケーションは、EDDB/IC・APIコールを行うが、ゲートウェイ・ライブラリは、そのQNX環境に対して一意的である「ネーム位置指定 (name locate)」、「送付 (send)」、「vcリリース (release)」のシステム・コールを、TCP/IPコマンド・パケットに変換し、そのパケットをゲートウェイに送る。QNXの下で走っているゲートウェイ・サービスは、次に、実際の「ネーム位置指定」、「送付」、「vcリリース」コールを提供してEDDBデータベースの情報にアクセスし、そのデータ及び状態を、同じ対のネットワーク・ソケット上をユーザ (クライアント) ・アプリケーション・プログラムに送り返す。すべてのパケット転送メカニズムは開発者には自明であり、あたかも、データがそのユーザに対して該ユーザの機械上でローカルに利用可能であるかのようにであることが明らかであろう。ゲートウェイ・ライブラリは、ANSI "C" で書かれており、開発者のターゲット・システムに利用可能にするためには最小のポーティング (porting) ・ワークを要求する。ゲートウェイ・サービスでは変更は全く不要であり、ゲートウェイ・サービスは、実行可能な形式でだけ利用可能であり、ゲートウェイPCが動いている限りゲートウェイ・メモリに関連して永久に動作する。ソフトウェアの観点からすると、ゲートウェイは、分散環境における、非標準的なTCP/IPプロトコル・アプリケーションである。

6

【0013】ゲートウェイの構成においては、ユーザは、通信要求を開始しかつサーバからの応答を待つ、第3の部分で構成するアプリケーションである。ユーザは、ユーザ・機械の中に「生きている」すなわち存在するものである。これと比較すると、サーバは、ゲートウェイの中に (永久に) 存在してユーザから来る通信要求を待つものである。サーバは、ユーザからの要求を受信した際には、必要なデータベース・アクセスを行い、その結果をユーザに返送する。本質的に、サーバは、ゲートウェイの中に存在して (生きて) おり、普遍的なデータベース・エンジンのように振る舞う。ゲートウェイは、データ・アイテムがユーザ (クライアント) ・アプリケーションによって結合されている限り、データベースからのダイナミックなイベント駆動データ更新をユーザに提供する。ユーザ・プログラムは、したがって、選択されたデータベース・アイテムにおいて実際の更新があった場合のみ、(最初にEDDB/ICデータベースによって告知された) サーバによって告知される。この付加的な要求によって、ゲートウェイ及びユーザ (クライアント) ・ノード上の1対の専用のソケットがコールすなわち呼び出され、(ゲートウェイからユーザへ接続する) 相互通信ができる。TCPトランスポート・プロトコルが用いられる。ゲートウェイは、複数のユーザ・ノードをEDDBデータベースに接続するサービスを提供する。各接続は、セッションを通じてユーザと共に確立されるそれ自身の一意的なソケットを有する。セッションは、ユーザによってなされる、要求情報へのゲートウェイの接続として定義される。セッションが終了すると、ソケットがそれ以降の使用のためにシステムに解放される。利用可能なソケットの数は、TCP/QNXデバイス・ドライバが提供できるものによって、制限されるだけである。

【0014】図4には、トップレベルのゲートウェイ・ソフトウェアを示しており、該ソフトウェアは、生座ライン・コンソール (PLC) とユーザのUNIX/VMX/Novellワークステーションとの間の接続をサポートするものである。ユーザ・アプリケーションは、EDDB/IC・APIコール (呼び出し) を行い、EDDB/ICデータベースを位置付け (位置指定) してそれを読み出す。ファンクション・コールは、データベース扱いとユーザ・アプリケーションに対するアイテム内容と共に返送される。ゲートウェイは、また、非同期イベント更新のためのユーザのタスクへの信号としてPLC又は検査接続から到来したEDDB/IC例外 (exceptions) をサポートする。たとえば、これらの外部イベントは、イベント・アングル変異パケット又は検査結果パケットである。このようにして、ゲートウェイ構成には、以下のような2つの主要データの流れがある。

(1) UNIX/VMS/Novellワークステーション

(5)

特開平7-367653

7

オンからPLCへのデータ取得

(2) 反対にPLCからUNIX/VMS/Novellワークステーションへの二次的なイベント更新

ゲートウェイは、すべての詳細な接続と、2つの異なるオペレーティング・システム (UNIX/VMS/Novell及びQNX) とネットワーク環境 (イーサネット上のTCP/IPと、ARCネット上のQNXネット) との間のデータ・パケット転送とを扱う。

【0015】ゲートウェイ構成は、図5に示されているように2つの部分で構成される。ゲートウェイ・ライブラリ (ユーザ・アプリケーション・ライブラリ) は、ユーザ・ノード上で生きている (使用可能な) 部分である。該ゲートウェイ・ライブラリは、ユーザ・アプリケーションとインターフェースして、イーサネット・ネットワーク上のTCPパケットを介して、ゲートウェイにおけるゲートウェイ・サービス (サーバ・アプリケーション・ソフトウェア) と会話を、ゲートウェイ・サービスは、ユーザ・アプリケーション・タスクの組で構成され、QNX環境において、EDDBデータベースを位置指定してアクセスする。サーバ・アプリケーションは、この結果をTCPパケットを介してクライアント・アプリケーションに返送する。この結果は、データベース問い合わせ (Inquiry) 又はEDDB例外に依存するものである。ゲートウェイ・ライブラリは、ユーザ・アプリケーションのプログラマがデータベースにアクセスするためにリンクさせる、ソフトウェアの層である。図6は、アプリケーションによって用いられるゲートウェイ・ライブラリの種々のモジュールの間のデータの流れを示している。

【0016】ユーザ・インターフェースのための関数は、以下のモジュールに含まれている。すなわち、ネーム位置指定すなわち位置付け (Name Locate)、送出 (Send)、及びイベント準備 (Prepare For Event) である。ユーザ (クライアント) ・コール・モジュールは、バクレー・ソケット・ライブラリにおける「送出」及び「受信」コールを行うことによって、TCPパケット転送を扱う。イベント準備の機能 (ファンクション) は、ユーザのノード上でのデータ転送のためのソケットを初期化し、別個のイベント・キュー・ソケットを確立することによって、ユーザ・イベント信号扱いメカニズムをセットアップする。ユーザがゲートウェイ・サービスとの接続を行った後で、ゲートウェイ・サービスは、他の必要な仕事を実行すると共に、それ自身のイベント・キュー・ソケットをセットアップし、そのユーザのイベント・キュー・ソケットへの接続を行うことを試みる。この接続が確立された場合には、EDDB/ICにおける付加されたデータベース・アイテム変更は、イベント・キュー・ソケットを介してユーザに送り戻される。メッセージはローカルにパッケージされ、ユーザからゲートウェイ・サーバに送ら

8

れ、そこで、パケットは、QNX「送出」メッセージの通過が行われる前に、バックが解かれデコードされる。受信されたパケットは、次に、EDDB/ICライブラリにおけるユーザ (クライアント) ・アプリケーション・タスク・バッファ空間に返送され、最終的には、ユーザ・アプリケーションに転送される。

【0017】図7は、ゲートウェイ・サービスとその外部のリソース (資源) との間のデータ・インターフェースに加えて、ゲートウェイ・サービス内のタスク間通信を表している。ゲートウェイ・サーバは、普遍的でセッション適用型のデータベース・サーバとして機能し、その構成は、以下の協調タスク (coordinating tasks) を備えている。

・「サーバ」タスクのメイン・プログラムは、ファイア・アップされたときに実行を開始し、永遠にランしつづける。該プログラムにより、プロトコル・ポートを取得し、プログラムをポート・マップ (mapper) に登録し、次に、データ・ソケットからのTCP/IPメッセージ・パケットとして到来するユーザ要求を受信するために待機 (リッスン) する。複数のユーザが一度に16のPLCまでアクセスできるようにしたことにより、メイン・プログラムは、チャネル上のデータが子 (child) 処理によって同時に処理される間に、より多くの入力接続を待たなければならない。作られる (spawned off) 予処理の組は、「コムズ (combs)」と「データ」とを含む。各セッション接続に1つである。これらのタスクは、QNXローカル・ネーム登録を介して相互に通信する。

【0018】通信タスクの「コムズ」が、最初にそれ自身を、ローカルなQNXクリアリングハウスに結び付ける。次に、永久実行 (ドワー・フォーエバー) ループ内に留まり、ブロッキング・モードの任意の入力データ・パケットを受信する。パケットを受信した後で、コムズは、最初に動作コードをチェックしてそれがポート・ナンバー (番号) パケットであるかどうかを見る。このポート・ナンバー・パケットは、ユーザによってセッションの確立処理の初期に送られて、ゲートウェイ・サービスに、データベース例外イベントにどのポートを送出すべきかを伝達する。このポート番号パケットによって、コムズがラッシュ・タスクを作ることになり、イベント・キュー・ソケット上のユーザに接続する。他のサポートされている動作コード (opcode) は、ネーム位置指定、送出、v cリリースである。これらのQNX動作の間の相違は、返答パケット・サイズにある。コムズ・タスクは、次に、パケットを、QNXメッセージ転送を介して、データ・タスクに送る。QNX送出コールはメッセージを受信することもあるので、送信コールが戻るとすぐに、プログラムは、返答TCPパケットを、TCP/IPネットワークを介してユーザに返送する。

9

【0019】・データベース・リンク・タスクの「データ」が、それ自身をローカルにその名前の下にあるクリアリングハウスと結び付けることによって開始する。次に、受信ブロック・ループに次の時まで永久に留まる。すなわち、コムズ・タスクが、ユーザ・アプリケーションからデータ・パケットを受信した後でそれにメッセージを送るまでか、又は、データベースに結び付いたアイテムに番号を与えるEDDB/IC例外が変更されるまでである。タスクは、コムズがメッセージ上を送られたために生じた場合には、動作コード（ネーム位置指定、送出、又はv cリリース）を最初にデコードし、それにしたがってQNX動作を実行する。次にタスクは、送り手のコムズ・タスクにその結果を返答する。タスクは、EDDB/ICデータベース例外番号を受信する場合には、QNXメッセージを直ちにラッシュ・タスクに送り、イベント・ソースを告知する。

【0020】・イベント告知タスクの「ラッシュ」は、コムズ・タスクによって作られ、最初にそれ自身をローカルにクリアリングハウスに結び付ける。次にそれは、例外イベント・キュー・ソケットをサーバにおいてイベント告知のためにセットアップする。イベント告知においては、ゲートウェイ・サーバは、実際には、ユーザのように振る舞い、それによって、それが作り出したソケットは接続をコールして、ユーザ（クライアント）・モード上のイベント・キュー・ソケットとの接続をしなければならない。次にタスクは、永久動作（ドゥー・フォーエバー）ループに入り、データ・タスクからデータを任意の例外イベントのために受信する。例外を受信した際には、次の2つのことを直に行う。すなわち、

1. 返答メッセージを、単にそのブロックを解く（un blocking）目的でデータ・タスクに送り戻す。
2. イベント告知パケットを、イベント・キュー・ソケットを介して、処理のためにユーザ・ノードに送る。

【0021】図8は、ユーザ（クライアント）接続要求に対する「リスニング・モード」において、ソケットを最初にセットアップするメイン・プログラムのためのプログラム構造図を示している。一旦タスクが永久動作ループ内に入り多くの接続を受け入れる場合には、到来する各接続は、結果的に、異なるソケット識別（id）と子タスク拡張数を生じることになる。これらの2つのパラメータは、次に、特定のセッションを与えるコムズ及びデータの子タスクの組を作るのに用いられる。複数のセッションがゲートウェイ内にオープン状態に開かれ得るので、コムズ又はデータの複数のコピーが同時にタスク・テーブル上に存在することになる。タスク間の通信は、したがって、タスク名ではなくタスク登録名に従って扱われなければならない。

【0022】図9は、別個のタスクとして構成された、ユーザからのパケットを処理し同じオープン状態のチャネル上で返答するタスクの「コムズ」のためのプログラ

(6)

特開平7-267653

10

ム構造図を示している。登録タスク・ネーム・セッションは、QNXシステム・コールの「ネーム付け」から成り、それ自身をローカルなクリアリングハウスに結び付けることによってデータ・タスクが通信できるようにする。準備段階の間は、コムズもまた、「ネーム位置付け（ネーム位置指定）」をコールすることによって対応する「データ」タスク名を位置指定しようとする。タスク名登録と位置指定とは、ソケットidと子タスク数とから成る同じコマンド・ライン・パラメータを用いることによってなされる。受信パケットは、ソケット接続が存在するユーザ・アプリケーションによって中断されるまで永久に継続すると考えられ、その場合には、コムズ・タスクはメッセージをデータ・タスクに送ってタスク自体が終了する前に存在させなければならない。これは、セッションが終了する場合にシステム内に残る「ソンド」処理を避けるために必要とされる。パケットを受信した後では、コムズ・タスクは、最初に動作コードをデコードし、それが「ネーム位置付け（ネーム位置指定）」、「送出」、又は「v c受信」であるかどうかを判断する。送出メッセージと返答パケットのサイズは異なっており、それに応じて識別特定されなければならない。プログラムは、次に、メッセージを処理のためにデータ・タスクに与え、その結果を取得し、それをユーザ・アプリケーションにネットワーク上を返却する。

【0023】図10は、コムズ・タスクからのメッセージ又はデータベース（EDDB又はC I）例外のどちらかを処理するデータ・タスクのためのプログラム構造図である。（例外ハンドラをセットアップするための）タスク登録及びイベント準備の後に、データベース・イベントが結び付けられたデータベース・アイテムから入力されるか、またはコムズ・タスクからのメッセージを取得するかまで、イベント取得ルーチンが受信をブロックすなわち阻止される。データベース・イベントを処理するために、データは、名前によってラッシュ・タスクを位置指定し、例外を位置指定されたものに送ろうとする。コムズから来たメッセージを処理するには、できる可能性がある。データは、最初に、コマンド・メッセージをデコードし、次に、それにしたがってQNXコールを実行する。この動作の結果は返答メッセージとしてコムズに送り返され、そのブロック（阻止）を解除する。コムズは、返答メッセージを取得するとすぐに、ユーザ・アプリケーションに戻るためにパケットを発生する。図11に示すようなラッシュ・タスクの場合には、データからの例外メッセージの受信を待ち、各メッセージをユーザ・アプリケーションにイベント・キュー・ソケット上を送り、データに返答してその送り手をブロック解除する。

【図面の簡単な説明】

【図1】本発明によるネットワーク・ゲートウェイとローカル・コンピュータと外部のアプリケーション環境と

11

の間の関係を示したシステム・アーキテクチャを示すブロック図である。

【図2】 QNX環境におけるイベント駆動型データベースのAPI構成を示すブロック図である。

【図3】 ネットワーク・ゲートウェイの分散型ソフトウェア構成を示すブロック図である。

【図4】 生産ライン・コンソールとホストとの間の接続をサポートするネットワーク・ゲートウェイ・ソフトウェアを示したトップレベル・コンテンツのデータ・フロー図である。

【図5】 ネットワーク・ゲートウェイに対する次のレベルのデータ・フロー図である。

【図6】 図5に示したゲートウェイ・ライブラリの種々

(7)

特開平7-267658

12

のモジュール間のデータの流れを示すデータ・フロー図である。

【図7】 図5に示したゲートウェイ・サービスとそのリソースとの間の、タスク間通信及びデータ・インターフェースを示すデータ・フロー図である。

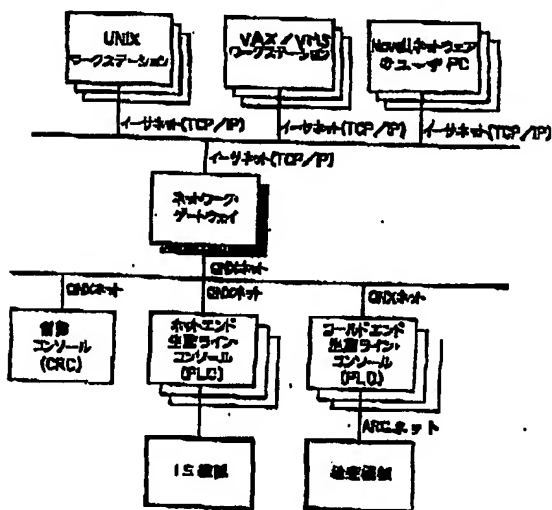
【図8】 ゲートウェイ・サービスに関するメイン・プログラム構造を示すブロック図である。

【図9】 ゲートウェイ通信タスク構造を示すブロック図である。

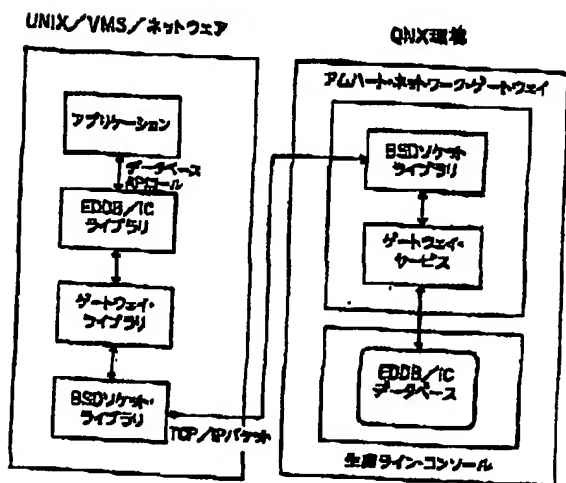
10 【図10】 ゲートウェイ・タスク・データ構造を示すブロック図である。

【図11】 ゲートウェイ・タスク・ラッシュ構造を示すブロック図である。

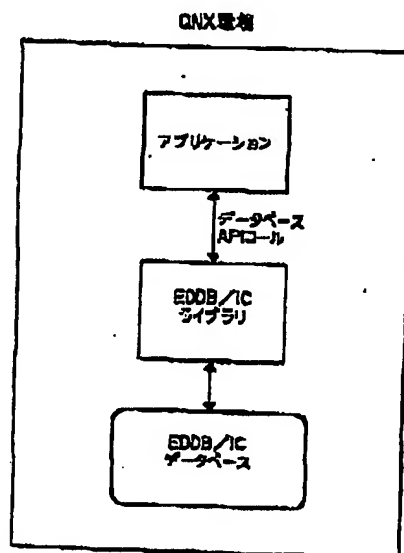
【図1】



【図3】



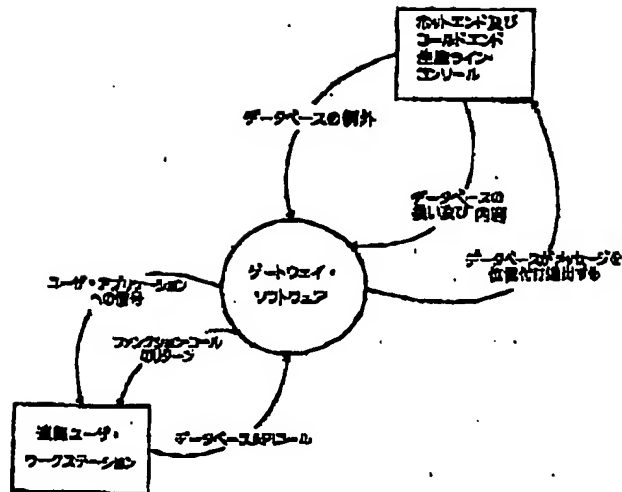
【図2】



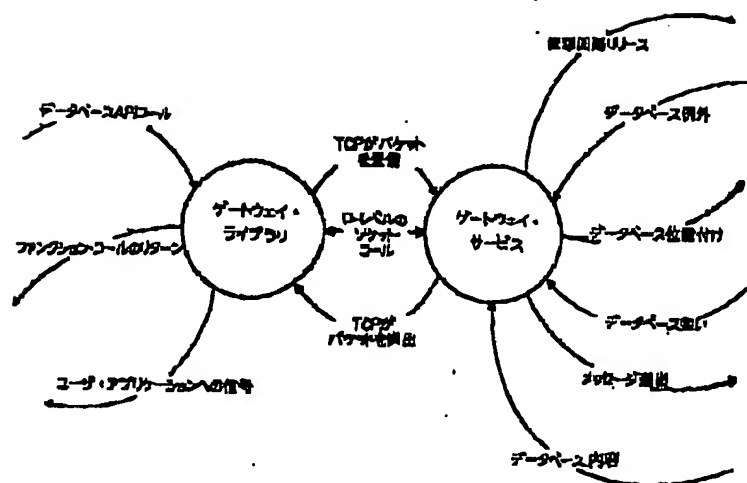
(8)

特開平7-267653

【図4】



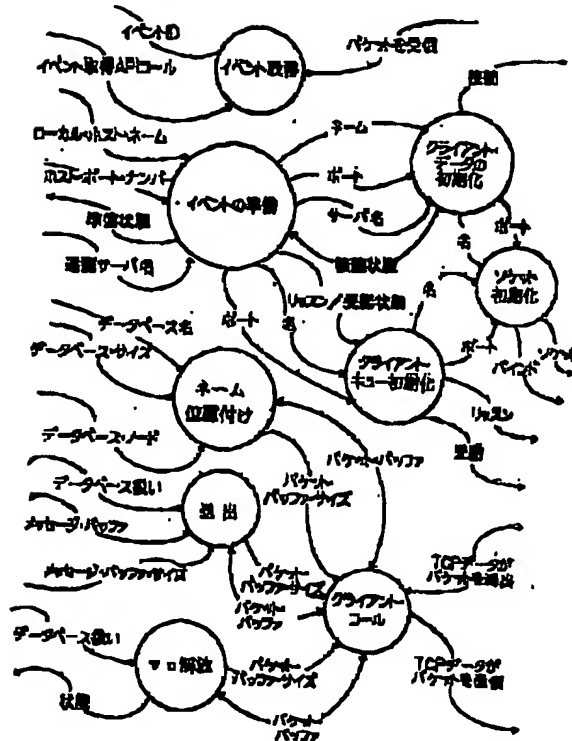
【図5】



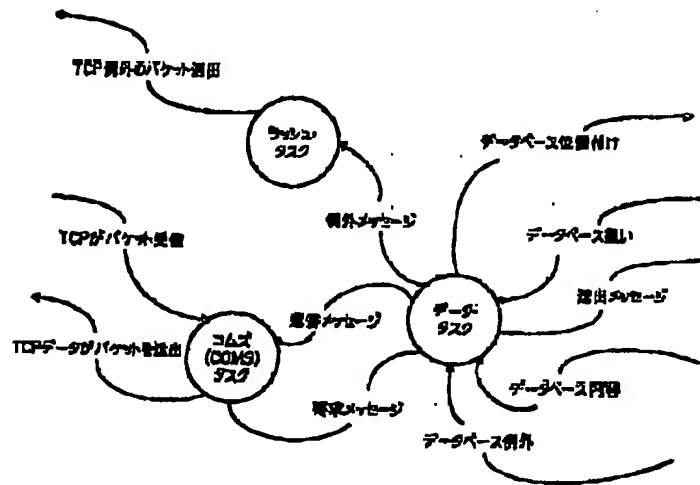
(9)

特開平7-267653

【図6】



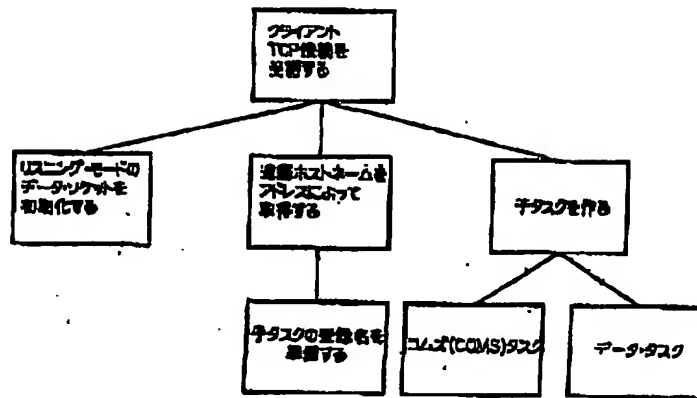
【図7】



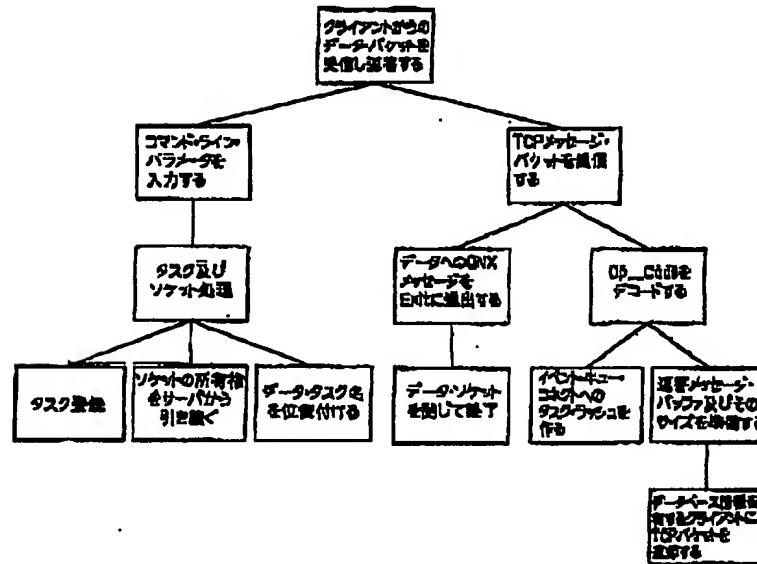
(10)

特開平7-267653

【図8】



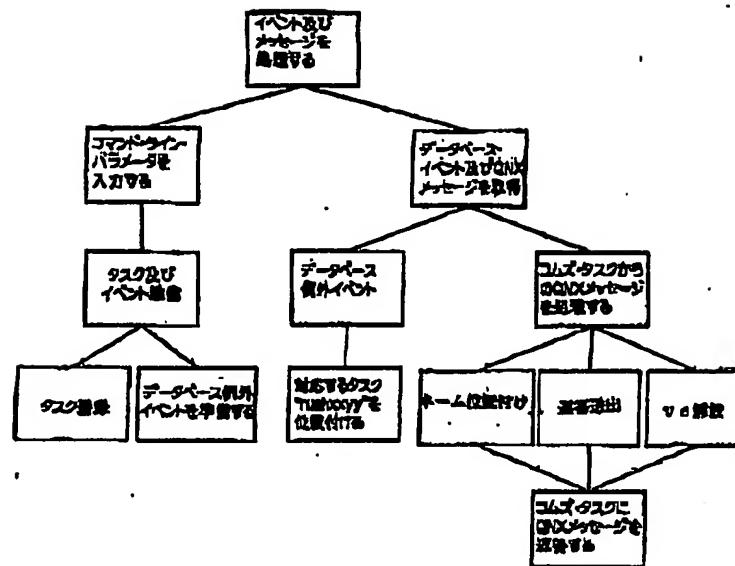
【図9】



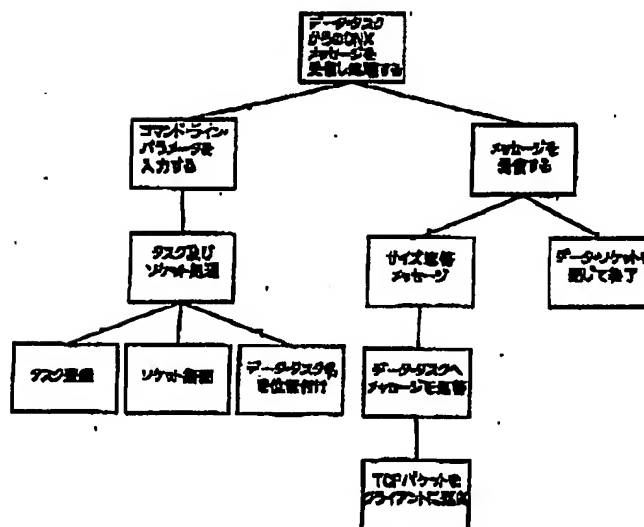
(11)

特開平7-267653

[10]



【圖 11】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☒ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.